BÁO CÁO THỰC HÀNH LAB 3

Thực hành môn Phương pháp học máy trong an toàn thông tin

Advanced Malware Detection

Nhóm: N07

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT101.M11.ANTN.1

STT	Họ và tên	MSSV	Email
1	Trần Hoàng Khang	19521671	19521671@gm.uit.edu.vn
2	Nguyễn Tú Ngọc	20521665	20521665@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:

STT	Công việc	Kết quả tự đánh giá
1	Câu hỏi 1	100%
2	Câu hỏi 2	100%
3	Câu hỏi 3	100%
4	Câu hỏi 4	100%
5	Câu hỏi 5	100%
6	Câu hỏi 6	100%
7	Câu hỏi 7	100%
8	Câu hỏi 8	0%

BÁO CÁO CHI TIẾT

Note: Giải thích được trình bày cụ thể trong file, được viết bằng English theo ý cá nhân

<More Practice, More Outstanding Result>

***1.** Cho biết kết quả accuracy và confusion matrix.*

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Kết quả:

```
y_test_pred = text_clf.predict(X_test)
print("Accuracy Score: %s" % accuracy_score(y_test, y_test_pred))
print("Confusion matrix: \n %s" % confusion_matrix(y_test, y_test_pred))
```

```
Accuracy Score: 0.9649910233393177
Confusion matrix:
 [[609  26]
 [ 13 466]]
```

2. Cho biết kết quả vector X

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Kết quả:

File thứ nhất:

```
['pdfDOCS_User_Reference_Guide-1.pdf', 'PythonBrochure.pdf']
/content/drive/MyDrive/Shared Drive/Lab3/Dataset/PDFSamples/
"/content/drive/MyDrive/Shared Drive/Lab3/Dataset/PDFSamples/pdfDOCS_User_Reference_Guide-1.pdf"
PDFiD 0.2.8 /content/drive/MyDrive/Shared Drive/Lab3/Dataset/PDFSamples/pdfDOCS_User_Reference_Guide-1.pdf
 PDF Header: %PDF-1.6
 obj                   153
 endobj                153
 stream                 82
 endstream              82
 xref                    2
 trailer                 2
 startxref               2
 /Page                   7
 /Encrypt                0
 /ObjStm                 0
 /JS                     0
 /JavaScript             0
 /AA                     0
 /OpenAction             0
 /AcroForm               2
 /JBIG2Decode            0
 /RichMedia              0
 /Launch                 0
 /EmbeddedFile           0
 /XFA                    0
 /Colors > 2^24          0
```

- File thứ 2 và 2 mảng số lượng thuộc tính trả về.

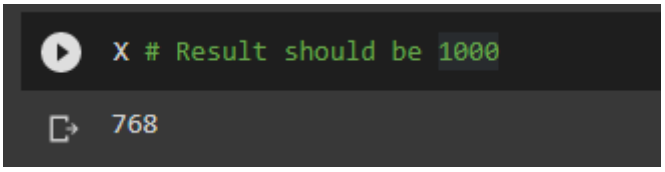
```
/content/drive/MyDrive/Shared Drive/Lab3/Dataset/PDFSamples/
"/content/drive/MyDrive/Shared Drive/Lab3/Dataset/PDFSamples/PythonBrochure.pdf"
PDFiD 0.2.8 /content/drive/MyDrive/Shared Drive/Lab3/Dataset/PDFSamples/PythonBrochure.pdf
 PDF Header: %PDF-1.6
 obj                  1096
 endobj               1095
 stream               1061
 endstream            1061
 xref                    0
 trailer                 0
 startxref               2
 /Page                  32
 /Encrypt                0
 /ObjStm                43
 /JS                     0
 /JavaScript             0
 /AA                     1
 /OpenAction             0
 /AcroForm               1
 /JBIG2Decode            0
 /RichMedia              0
 /Launch                 0
 /EmbeddedFile           0
 /XFA                    0
 /Colors > 2^24          0


[[153, 153, 82, 82, 2, 2, 2, 7, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0], [1096, 1095, 1061, 1061, 0, 0, 2, 32, 0, 43, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0]]
```

***3.** Cho biết kết quả vector X*

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Kết quả:

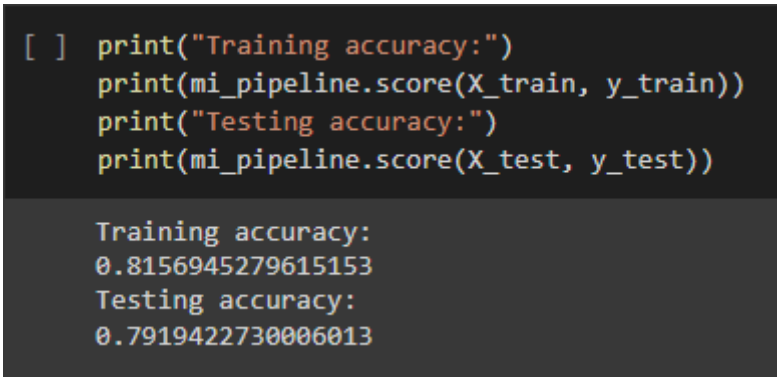


Note: Như comment, số lượng trả về phải là *1000*. Phần này chưa thực hiện được.

***4.** Cho biết kết quả đánh giá*

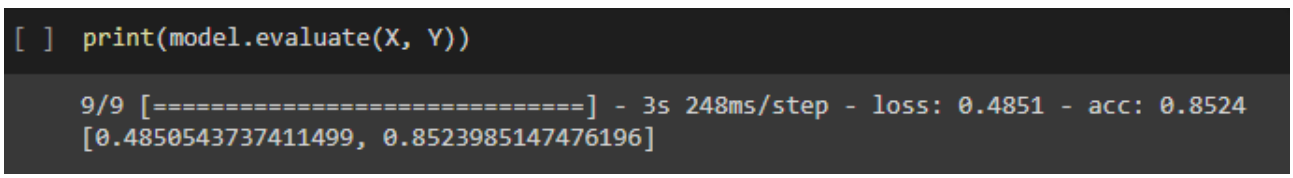
<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Kết quả evalation:



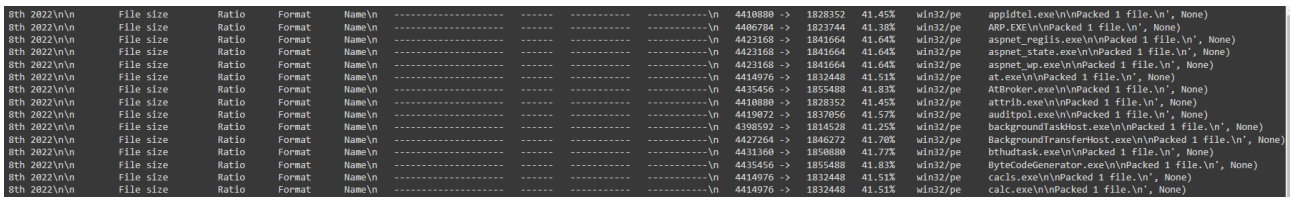
5. *Cho biết kết quả đánh giá mô hình qua tập test.*

<Xem kết quả chi tiết tại file Notebook (.ipynb)>



6. *Cài đặt. UPX từ https://github.com/1.upx/upx/releases, và tiến hành đóng gói các tập tin pe tại Benign PE Samples UPX*

Kết quả chạy khi packed thành công:



7. *Cho biết kết quả đánh giá.*

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

```
from sklearn.metrics import confusion_matrix, accuracy_score, recall_score, f1_score, precision_score, roc_auc_score

print("Confusion matrix:\n %s" % confusion_matrix(y_test, y_pred))

# Additional metrics (Not required)
print("Accurcy: %s" %(accuracy_score(y_test, y_pred)))
print("Recall: %s" %(recall_score(y_test, y_pred, average='micro')))
print("Precision: %s" %(precision_score(y_test, y_pred, average='micro')))
print("F1 Score: %s" %(f1_score(y_test, y_pred, average='micro')))
print("ROC AUC: %s" %(roc_auc_score(y_test, y_pred)))
```

```
Confusion matrix:
 [[69  0]
 [ 0 23]]
Accurcy: 1.0
Recall: 1.0
Precision: 1.0
F1 Score: 1.0
ROC AUC: 1.0
```

8. *Cho biết kết quả đánh giá mẫu mới trong việc đánh lừa bộ nhận diện.*

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Hiện tại đang gặp lỗi khi chạy model với các tài nguyên file **.py** cho trước. Sẽ fix trong tương lai

Bonus round: *Thực hiện chạy và giải thích model từ trang Your First Deep Learning Project in Python with Keras Step-by-Step (machinelearningmastery.com).*

*- Áp dụng GridSearch với **model cho trước***

*- Áp dụng GridSearch với **DecisionTree Classifier***

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Kết quả chạy model với tham số được định nghĩa trong blog:

```
[19] # evaluate the keras model
     _, accuracy = model.evaluate(X, y)
     print('Accuracy: %.2f' % (accuracy*100))

     24/24 [==============================] - 0s 2ms/step - loss: 0.5154 - accuracy: 0.7474
     Accuracy: 74.74
```

Kết quả khi áp dụng *GridSearch* với các tham số trong mô hình huấn luyện (sử dụng model như trên):

```
params_dict = {
    "criterion" : ['gini','entropy'],
    "max_depth" : [1,2,3,4,5,6,7,None]
}

gs = GridSearchCV(dt,param_grid = params_dict,cv = 10)
gs.fit(x_train,y_train)

GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [1, 2, 3, 4, 5, 6, 7, None]})
```

```
print(best_params)
print(accuracy)

{'batch_size': 20, 'nb_epoch': 400, 'unit': 11}
0.7343130469322204
```

Kết quả khi áp dụng *GridSearch* với các tham số khi cài đặt thuật toán ***Decision Tree:***

```
params_dict = {
    "criterion" : ['gini','entropy'],
    "max_depth" : [1,2,3,4,5,6,7,None]
}

gs = GridSearchCV(dt,param_grid = params_dict,cv = 10)
gs.fit(x_train,y_train)

GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [1, 2, 3, 4, 5, 6, 7, None]})
```

```
[33] print(gs.best_params_)

     {'criterion': 'gini', 'max_depth': 5}
```